Willow System Demonstration

John C. Knight Jonathan Hill Philip Varner
Department of Computer Science
University of Virginia
{knight|jch8f|varner}@cs.virginia.edu

Alexander L. Wolf Dennis Heimbigner
Department of Computer Science
University of Colorado
{alw|dennis}@cs.colorado.edu

Premkumar Devanbu
Department of Computer Science
University of California, Davis
devanbu@cs.ucdavis.edu

Abstract

Dealing with damage that arises during operation of networked information systems is essential if such systems are to provide the dependability required by modern critical applications. Extensive damage can arise from environmental factors, malicious actions and so on, and in most cases it is impractical to mask the effects of such damage using typical redundancy techniques. Reconfiguration is required of both the application and the underlying computing and communications fabric. Such reconfiguration is difficult to achieve because it requires communication with a significant number of nodes both to determine the problem and to effect a repair. In this demonstration we present an approach to the implementation of such reconfiguration. The approach to reactive control includes formal description of the error states, synthesis of the implementation, a novel new communications mechanism for communication between the error detection system and the application, and a system for coordinating the effects of independent actions.

1. System overview

As a society, we are becoming increasingly dependent on the continuous, proper functioning of large-scale, heterogeneous, distributed information systems. These systems are formed from large numbers of components, both hardware and software, originating from multiple sources assembled into complex and evolving structures spread across wide geographic areas. In view of their importance, it is desirable to make these systems *survivable*, i.e., to ensure that these systems can either avoid disruptions or can continue to provide acceptable service, though not necessarily complete levels of service, in the face of serious disruptions to their normal operation [1]. This demonstration will be of an architecture, the Willow architecture, designed to provide these capabilities. The types of disruption with which we are concerned are physical damage from terrorists, widespread environmental damage, coordinated security attacks, hardware failures, operational errors, etc.

The Willow architecture is based on the notion that survivability of a distributed application requires a broad approach to dealing with faults in the application, an approach that includes fault avoidance, fault elimination, and fault tolerance. Thus it includes mechanisms: (a) to *avoid* the introduction of faults into the systems at the time of initial deployment or subsequent enhancement; (b) to *eliminate* (i.e., remove) faults from a deployed application once they are either identified or merely suspected but before they can cause failure; and (c) to *tolerate* the effects of faults during operation. These various mechanisms are all based on a general notion of *reconfiguration* at both the system and the application levels together with a framework that implements a monitor/analyze/respond approach to the identification and treatment of faults.

The approach to the implementation of survivability is to supplement the usual information system fabric with a comprehensive fault-tolerance mechanism that is referred to as a *survivability architecture* or an *information survivability control system* [2]. This mechanism consists of two major parts: (1) an error detection component to diagnose system state and determine what if anything needs to be done; and (2) an actuation component that effects changes to both the distributed application and the underlying computing infrastructure. The error-detection mechanism is designed to correlate damage information in both space and time, and to permit sequences of corrective actions to be taken if necessary should damage become worse over time. The error states of interest are defined in a specialized formal language and implementations of error detection mechanisms are synthesized from specifications in that language. The actuation mechanism is designed to permit corrective action to be applied to relevant elements of the system efficiently and with minimal implementation effort.

Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX'03)
0-7695-1897-4/03 $17.00 © 2003 IEEE

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2003		**00-00-2003 to 00-00-2003**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Willow System Demonstration	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Virginia,Department of Computer Science,151 Engineer's Way,Charlottesville,VA,22904-4740**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**3**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

A central difficulty that the designer of such an architecture faces is scalability. The information systems of interest are composed typically of very large numbers of computing nodes and extensive communications facilities. The damage that arises from a disruption can be very complex, and the number of actions needed to achieve a suitable recovery from damage can involve many system components. The implementation strategy that Willow uses is designed to deal with precisely these issues of scale.

The most direct approach to error detection and recovery is to base analysis on a model of the information system's dynamic state stored in some centralized repository such as a database. State here includes information such as what computing nodes are being used for what function at any given time, details of the operational hardware and its characteristics, the operational communications topology, and so on. Such an approach is very difficult for two reasons: (1) accessing state information for a large distributed information system from a database becomes more and more difficult as the size increases and eventually becomes infeasible; and (2) the model of the state is bound to differ from the true state to some extent because of the inevitable delay in recording changes. The first of these two reasons limits the size of systems that can be accommodated, and the second limits the treatment of systems with dynamic membership such as wireless networks.

The design approach of the Willow architecture is to avoid the use of centralized dynamic state information to the extent possible. The error detection component operates by capturing state information of damaged system components only as damage is detected. It discards the information once the damage has been treated. The error recovery (or actuation) component operates using *intentional* addressing, i.e., system components are addressed based on their current state rather than with some synthetic address such as an IP number. Thus the representation of the dynamic state is distributed throughout the system. Combined, these two approaches allow a wide variety of faults to be dealt with efficiently even for very large networks.

2. Application to general terrorist attacks

The Willow architecture can be used for systems other than distributed information networks. Many existing infrastructure systems are vulnerable to coordinated attacks by terrorists. If a terrorist group were to undertake a number of small, separate actions at approximately the same time, i.e., a coordinated attack, the combined effect could be catastrophic even though no single action was very serious. As an example, consider the possibility of terrorists attempting to breach airport security. If a few terrorists attempt to smuggle weapons at any single airport, most will be caught but one or two might succeed. To the security staff, those caught might look like a statistical anomaly. However, if such "anomalies" were to arise in the same time frame at *several* airports, it would almost certainly indicate a coordinated action with the potential serious consequence of having a significant number of armed terrorists aboard aircraft. Such a circumstance could only be detected if the suspicious "anomalies" at the various airports involved were known about and analyzed as a group in real time. Other similar vulnerabilities abound. For example, disruption of operations at an electric generation facility resulting from detonation of a truck bomb, though serious, would not be catastrophic. Several such attacks at the same time but at different locations would almost certainly be catastrophic.

The Willow technology can be applied to terrorist attacks very effectively. A system could collect information about incidents, analyze that information to detect possible attacks, present the results of the analysis to law-enforcement officials, and, either allow those officials to disseminate appropriate responses or to disseminate predefined responses automatically. All of these steps would be achieved in short time frames so as to maximize the chances of minimizing the effects of an attack.

The form of such a system is to equip security staff with small hand-held computers connected to the Internet. When an event occurs that might be a symptom of an attack, the security staff indicate the event using a very simple interface on the computer. Separate computers performing analysis would collect such events from all sources and make a determination about whether the observed events indicate a coordinated attack. Display of the observed events for law-enforcement officials is clearly desirable and simple to achieve. Once an attack is detected, the system would communicate with the hand-held computers in order to indicate any necessary immediate response. Responses might include termination of operations, revised security procedures, introduction of law-enforcement officials, and so on. For airport security, security staff would indicate when a banned object was found using a touch screen. Patterns of events known to be characteristic of serious attacks, either regional or national, would be detected and displayed. Optionally, necessary responses could be communicated to the airport security staff.

In the Willow architecture, the various components and all of the algorithms that they use have been designed to scale to very large information networks. It will scale easily to the sizes necessary for essentially any attack-detection system. If all transportation facilities, energy-production facilities, water-treatment facilities, public buildings and so on were equipped with the type of hand-

held device mentioned above, it is likely that the number of such devices would be of the order of hundreds of thousands or higher. This is comparable to the information network target for Willow.

The Willow terrorist attack detection and response system has the following six major components:

1. Software within the hand-held computers that generates an event message whenever a user indicates an incident via the touch screen.
2. The TEDL specification language that is used to define the circumstances that are thought to constitute an attack. The use of a specification language permits the circumstances defined to constitute an attack to be changed easily as new information about terrorist threats is acquired.
3. The TEDL translator that creates analysis software from a specification written in the TEDL language. This translator ensures that the software needed for analysis can be created quickly and easily, and that the need to change is accommodated by synthesis of the analysis software.
4. A site-select addressing mechanism used to address sets of nodes to transmit response information. This mechanism has certain limitations but the major advantage that it provides is the ability to address sets of nodes without knowing specific machine names or addresses.
5. A mechanism for coordinating the effects of multiple responses that might conflict. This component is needed because separate, independent analysis might be required to cope with different anticipated threats. Thus separate conflicting responses might be attempted at the same time and this cannot be permitted.
6. The Siena publish/subscribe information system. This system provides the necessary communications infrastructure upon which all the other components depend.

This demonstration will be a Willow system operating on a target network consisting of 120 computers, each supporting several nodes. Distributed application running on this target system will be shown, and faults modeling a variety of forms of failure, damage, and attack will be introduced. The error detection and recovery mechanisms in the Willow survivability architecture will be demonstrated. Part of the demonstration will be of the system designed to detect and respond to widely dispersed and coordinated terrorist attacks.

Acknowledgments

This work was supported in part by the Defense Advanced Research Projects Agency under grant N66001-00-8945 (SPAWAR), the Air Force Research Laboratory under grant F30602-01-1-0503, and Microsoft Corporation. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of any of the sponsors.

References

[1] Strunk, E.A., J.C. Knight, and K.J. Sullivan, “Towards a rigorous definition of information system survivability”, Proceedings of DISCEX 3, Third DARPA Information Survivability Conference and Exposition, Washington D.C., April 2003.

[2] Sullivan, K.J., J.C. Knight, X. Du, and S. Geist, “Information Survivability Control Systems”, Proceedings of ICSE 21: Twenty First International Conference on Software Engineering, Los Angeles, CA, May 1999.